ELSEVIER

Computer Networks 31 (1999) 2227–2236

XP-002185729

P.D. 10-11-1999

P. 2227-2236 (10)

COMPUTER NETWORKS

www.elsevier.com/locate/comnet

C1

New approach for management services with a Web browser

Maciej Jarkowski [*], Artur Binczewski [1], Maciej Stroinski [2]

Poznan Supercomputing and Networking Center (PSNC), Noskowskiego 10, 61-704 Poznań, Poland

Abstract

This paper presents a new approach for integrated management services, which depend on the use of a Web browser for access to MIB objects. This approach allows the user to perform network management from any host on the network, guarantees the safety of transmissions between the Web browser and the server and permits the visualization of variables of the MIB in different ways. The most interesting feature is the possibility for the user to define the appearance of the management interface. The system, called *WebMan*, was implemented at the Poznań Supercomputing and Networking Center (PSNC) and it is now in production for management of its UPS devices. © 1999 Elsevier Science B.V. All rights reserved.

Keywords: Computer networks; Management; Web browser; MIB; Security

1. Introduction

Over the past few years there has been a progressive development of computer technology. This is the result of the human's natural need for more working comfort and desire to find new technological solutions. The quick development of computer infrastructures, and thereby computer networks, also creates the necessity to look for new, more efficient and more effective methods for maintaining the network. This is caused by the growing requirements put on modern computer networks such as available bandwidth, security and reliability of the services provided. Effective network operation is only possible when management systems form an integral part of the production network.

Today computer networks are a complex system consisting of many interconnected networks. The integral part of each network should be, and is, the management. The management of networks is necessary for:
- Management of configuration,
- Management of performance,
- Management of security,
- Management of billing,
- Management of faults.

The most commonly implemented management models are based on the SNMP protocol. The agents in most management systems are integrated into the system or are standalone management applications for use with equipment from a particular supplier. In the presence of the heterogeneous structure of the

* Corresponding author. E-mail: Maciej.Jarkowski@pl.ibm.com

[1] Tel.: +48-61-858-2010; fax: +48-61-852-5954; e-mail: artur@man.poznan.pl

[2] Tel.: +48-61-858-2003; fax: +48-61-852-5954; e-mail: stroins@man.poznan.pl

1389-1286/99/$ - see front matter
PII: S1389-1286(99)00098-5

network the most common approach is management of the equipment from a single management platform (for example the Solstice Enterprise Manager by SunSoft, the NetView 6000 by IBM and the HP Open View by Hewlett Packard) with many applications which each manage the defined objects specified in the RFC MIBs or in the system's private MIBs. In this situation, management is a difficult process largely based on centralized capacity. One way to simplify the process is to use a Web browser as the GUI for the management system.

In many cases an administrator, who is responsible for a specific subnet or virtual network, requires specific access to management objects relating to these networks. In this situation the problem could be resolved by offering the users/administrator secured access to a range of objects specified by that user, via a Web browser. This concept of management services could also be used for a wide range of control functions of specific domains within the network. In this case one management centre could offer many services to the distributed domain managers.

If we look at the problem from this perspective we see that management services, with the possibility of defining objects by the users, considerably extend the functionality of management services by using a Web browser to access commercially available modules on a central management system. A pilot system, based on this concept, was designed and implemented by the Poznań Supercomputing and Networking Center to offer management services to the Poznań Metropolitan Area Network and the Polish national academic network POL-34.

2. Problem statement

Recently we have seen the complete integration of management systems using a Web interface as a GUI for access to the system. Management systems using the Web are very convenient, because the administrator can use the management software from anywhere he wants. He can even use a modem and a dial-up connection to manage his devices as the required software is automatically downloaded from the server by the Web browser.

An example of such a system is the Dr-Web series from SNMP Research International (Dr-Web Manager and Dr-Web Agent). These applications provide an interface between the SNMP agent and the Web browser and allow access to selected SNMP variables by specific URLs.

Other types of the Web management software include applications, which have a similar function to regular management software, but with a graphical Web interface. Thus, they have a great advantage over traditional systems as they allow the network to be managed from anywhere the administrator can connect to the TCP/IP network. These systems can perform an automatic discovery of the network, build a logical network map, visualize the status of the network and present the collected performance data in graphical form. However, as such management applications are a remote interface to the management system itself, the software must have some security features, like encrypted communication between server and client and user identifications. This is very important because access to the management system from any host on the network may help somebody to intercept data of crucial importance for the life of the network. Examples of such systems are IntraSpection from Asanté Technologies or NetView 5 from IBM.

The advantage of the software described above is the graphical visualization of the managed devices and the connections between them. Additionally, Web interface applications are accessible from any host on the connected TCP/IP network, and this is very convenient for the administrators.

Unfortunately, the systems available on the market can only visualize SNMP variables gathered from various SNMP agents in a very simple way, showing the SNMP variables as text or number. The systems do not allow users to create their own management panels.

The system developed by the PSNC (*WebMan*) resolves the problems described above and is presented in the next section.

3. The WebMan system architecture

The WebMan system allows the management of devices and their access to SNMP agent variables and then follows the access of specific parameters

available in a private MIB-like status of the ATM links, battery temperature in the UPS, error condition, and so on.

This application, called *WebMan* (short for *Web Management*), consists of two parts — server and client. The Web browser runs the client and the server runs on the management stations. The client application is a Java applet, which helps the administrator with managing the visualization of the managed SNMP objects. The server is the interface between the Web browser and the management station that has access to all managed devices. The server communicates with the client with two protocols: HTTP is used for browser requests and another non-standard protocol is used for communicating between the applet and the server. This protocol is designed for remote API function calling. It is very simple (easier than RFC or CORBA), and its commands are translated by the server to the management system API calls. As a transport layer it uses TCP/IP (similar to HTTP), so the connections are reliable. To increase security in connections and communication both protocols work on the same TCP port. This means that the client communicates with a server using only one TCP/IP port and then the server waits for connections on one TCP/IP port. In addition, every suspicious connection or message causes a break of connection by the server. The server with the management system communicates with the management system API functions.

The client process can visualize managed objects using the Web interface. The client allows the administrator to build his own management console (panel) using predefined objects that visualize the SNMP variables. These objects are as follows:

- Diode: this can be in a variety of predefined colours in relation to the SNMP object value.
- Bar: its height represents the value of an SNMP object.
- Switch: this represents the name of the value of SNMP integer objects and also works like a switch.
- Text box: this represents SNMP object values in a textual form depending on the type of object.
- Table: this can represent an SNMP table.
- Chart: this shows a value as a time chart.
- Trap object: this catches all traps from a defined IP address and then triggers an alarm. A user can browse all received traps up to the values of embedded trap variables.
- Subpanel: this allows hierarchical bounded panels to be built.

Additionally there are some helpful options, such as labels, that facilitate the use of the system. Users can also make use of some predefined objects (templates). This makes the system easier to use for people who do not know the SNMP protocol. Each object representing an integer value can also trigger an alarm when it reaches a certain defined value.

For optimal efficiency in building the management panel a special, easy to use, editor was designed. The editor is very intuitive, like the visual dialog editor included in Microsoft Visual C. It allows the user to place an object in the panel, change its visual attributes (i.e. size and position) and modify specific parameters. It also has the ability to copy, cut and paste objects to and from an internal clipboard.

All available objects are included in a dedicated window called the *Toolbar* (Fig. 1). To assist the less experienced user to use the WebMan management panel editor, it is possible to create partly defined objects called templates. This speeds up and simplifies the process of defining panels, because the end-user does not have to know the SNMP protocol or the OID of the requested SNMP variable.

With this functionality the user can easily define his own management panel which visualizes important objects included in the SNMP agents of the managed devices. This leads to the simplification of management because the administrator builds the management environment himself according to his inclination, habit and wishes.

As a management system that makes its services available via a Web interface, NetView 6000 version

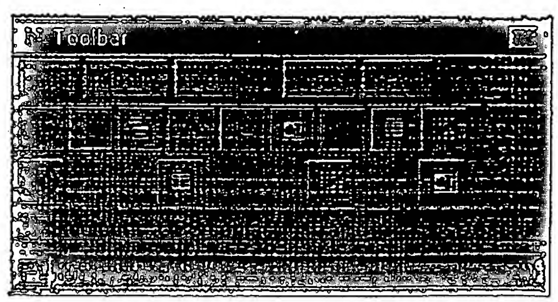

Fig. 1. The Toolbar window.

4 was chosen. It offers great management abilities and a complex API which supports SNMP services. It easily allows the implementation of functions that gather data from SNMP agents and sets their variables with new values. There are also NetView database and NetView maps used with this API. However, it should be noted that WebMan could easily be implemented on other systems that have a similar functionality to NetView.

The WebMan server is located on a machine running the management system. There are some advantages to this solution such as:

- There is a management system database and maps used for storing WebMan objects, parameters and users attributes.
- The server uses a management system API, which simplifies the use of the SNMP protocol and allows access to management system databases and maps.
- All management devices send traps to the management system, so WebMan can catch all of these traps.
- Many devices can be accessed only from management stations, so WebMan can access all available devices. There is full access to all managed devices.

This system also has some security features such as user authentication and transmission encoding. These features are performed by the fast stream algorithm (RC4), which uses symmetric keys.

Many users can be defined in the system. The system administrator can set different levels of authorization allowing clients to access only some of the devices. The security includes limiting access to the devices based on IP addresses and permitting read–write or read-only operations.

Because the centre point of this system is the server running on the management station, all configuration data is stored in a NetView database. This information includes all users' rights and management panels. For storing this data there are specially defined database fields, which are also attributes of NetView objects. Therefore, all user attributes are stored in the objects representing each user's home map on a NetView submap. All other objects generated by WebMan are represented by symbols on the hierarchical NetView submaps below the user submap. Each object on the management panel has

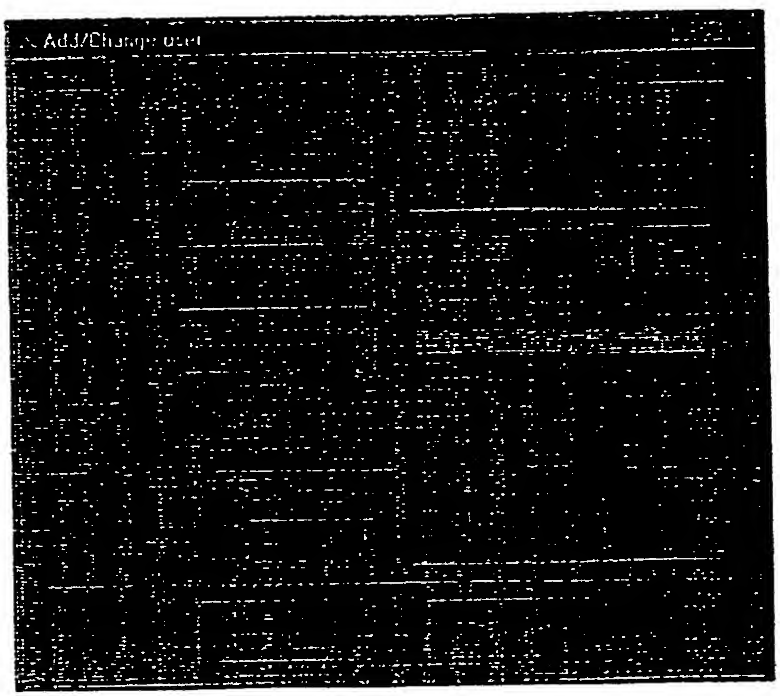

Fig. 2. Dialog window for WebMan user's administration.

its mirror in the NetView database. In the root map there is a symbol representing the WebMan submap that includes its users' individual objects. Each submap in the WebMan user submap tree represents one management panel.

NetView also allows simple integration with a user's own application. Thus, all user management functions are performed from within NetView menus using additional graphical applications (Fig. 2). This is very helpful for administrating the WebMan system.

WebMan could be a very interesting application for commercial ISPs (Internet Service Providers). Such companies often have their own management station for controlling their networks. If an ISP used WebMan they can also sell management services to their end-users. End-users could use WebMan (provided by the ISP) for managing their own devices. The end-user could define suitable management panels that fulfil his requirements for network and device controlling. This would be a much cheaper solution than buying a management station and software of his own.

Fig. 3 shows the architecture of WebMan. The server consists of two modules. The first is responsible for HTTP requests. The second is the main WebMan server that makes management services available on the Web. Between the two servers there is a multiplexer module which directs received requests from the network to the appropriate server.

The Web server knows the set of files that are available from the HTTP requests. These are precompiled Java classes that form the client applet and the main HTML page that run the client applet.

In the main WebMan server there is also a decoding module. Above it there is a module that services the WebMan protocol requests. It translates the client requests to a sequence of NetView API calls. However, before the calls are executed the system checks that the user has the rights to execute those particular functions.

Thus, WebMan also has two clients. The first is the ordinary Web browser that supports Java 1.1 applets. The second is this applet. It has (as the WebMan server) a decoding module and a module that services WebMan protocol requests. Above it

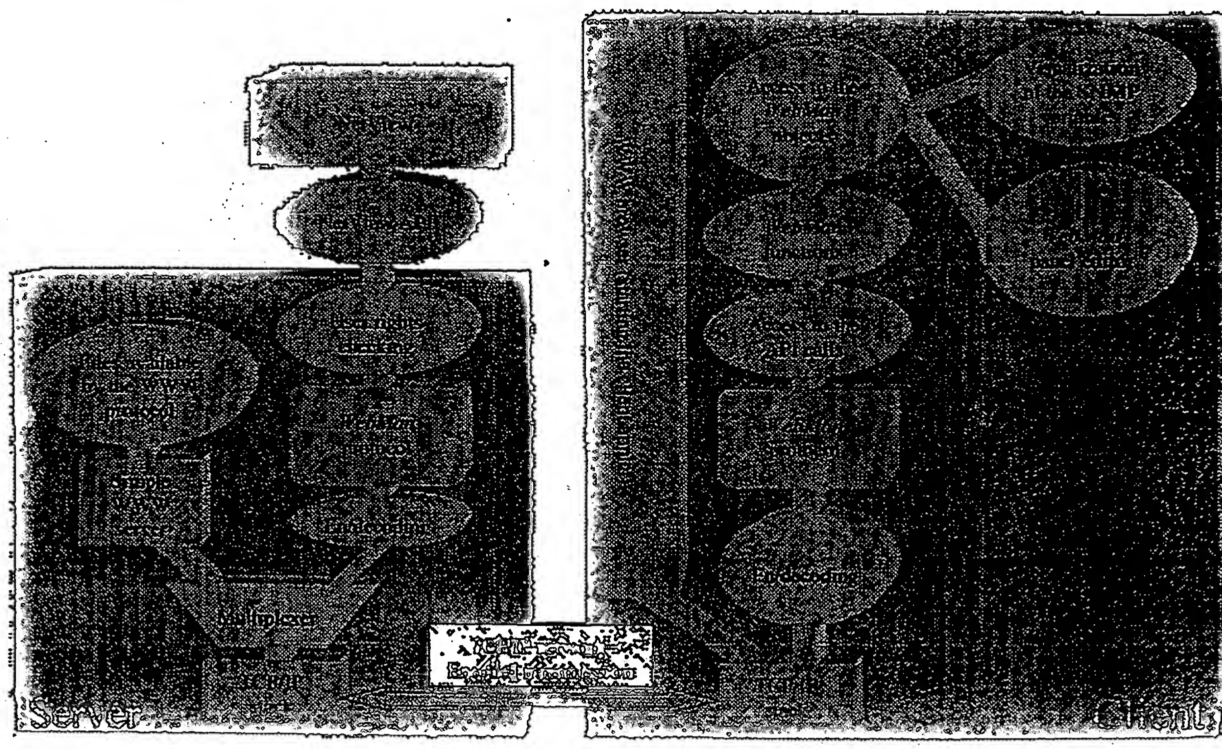

Fig. 3. The architecture of the WebMan.

there are modules that implement remote access to the NetView functions from the WebMan objects. The client can also visualize SNMP variables and has an editor for creating management panels. Both these functions require access to the WebMan objects defined in the NetView database. This is enabled with WebMan functions, which remotely execute NetView calls.

The project of this system is expected to run its server on the same station as the whole management system. This solution may decrease the system performance depending on the number of managed objects, available operational memory, the storage device capacity and the CPU (Central Processing Unit) speed. This is a reason to consider another implementation of the WebMan server.

It is planned that there will be another dedicated station to the WebMan server, only. The second station must run the whole NetView system, but it does nothing but service the WebMan server. The only additional requirement will be the receiving of all traps from the main management station and access to all managed devices.

This solution lightens the load of the main management station, because although the server itself does not load the CPU and the memory very much, the most critical operations are those that perform database accesses and these require a lot of memory. Therefore entering many objects (using the WebMan system) to the NetView database may be very critical for the performance of the management station. In this situation the dedicated WebMan station will be a good solution.

When there are many periodically gathered data from SNMP agents, there may be increased network traffic between the server and client. Of course, this depends on the amount of gathered data and on the frequency of gathering. Note that the main management station generates some traffic that also depends on the amount of data. So, if the WebMan server is on the same station as the NetView system, then the network traffic can also be increased.

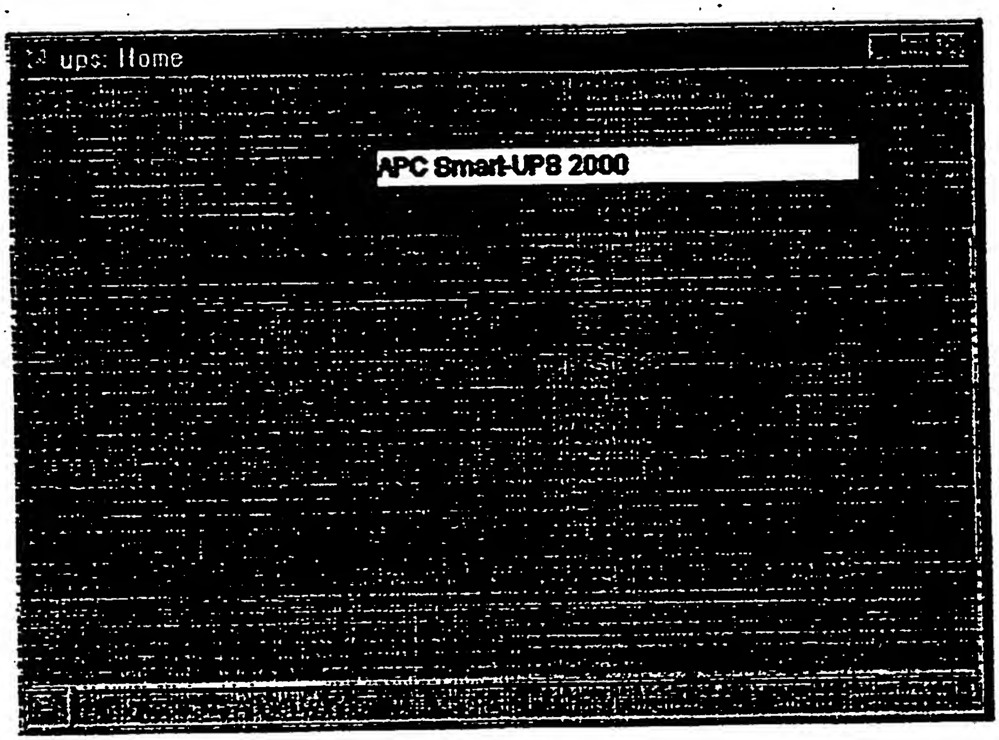

Fig. 4. Main management panel for the UPS.

The application was tested at the Poznań Supercomputing and Networking Center (PSNC) on an IBM RS6000 workstation with 96 MB of RAM and a PowerPC CPU. The NetView 6000 v4.1 was installed on an AIX 4.1.4. The client was run in Netscape Communicator 4.05 on a Pentium 166 PC with 32 MB of RAM running Windows 95. Section 4 provides an example of the WebMan system in use.

4. Management of the UPSs by the WebMan

At PSNC an application for the management of the UPS system was implemented using WebMan. There are management panels for controlling UPSs from APC (American Power Conversion). In Fig. 4, the main management panel is intended for the management of the UPS. Note that this is only an example. The UPS agent from APC has more variables than are used in this example, but the ones shown are the most important and illustrate the ability of the WebMan system. Most of the objects have labels, that are shown after a half-second keeping the mouse above the object. These labels describe the meaning of the object.

The main panel shows the identification data of the UPS (IP address and the model of the UPS gathered from the private APC MIB). In the middle there are objects presenting the current status of the UPS. A green coloured diode shows the correct status (UPS is in the on-line state in Fig. 4). This allows for fast UPS status recognition. If the UPS goes off-line this diode becomes red and an alarm is sounded. Near the diode there is the switch which shows, in a text form, the status of the UPS. These

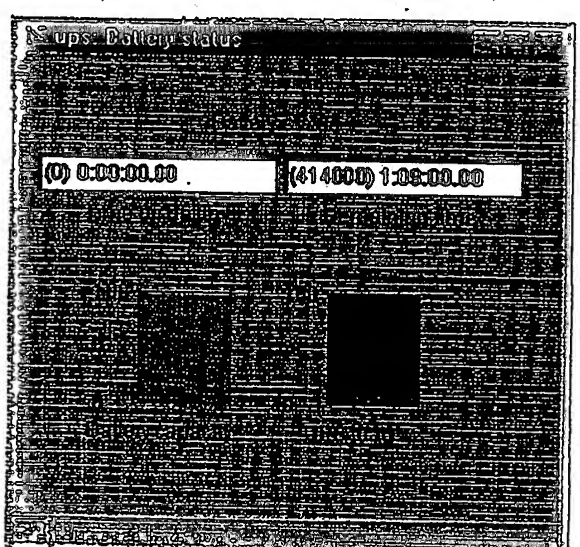

Fig. 5. The battery status panel.

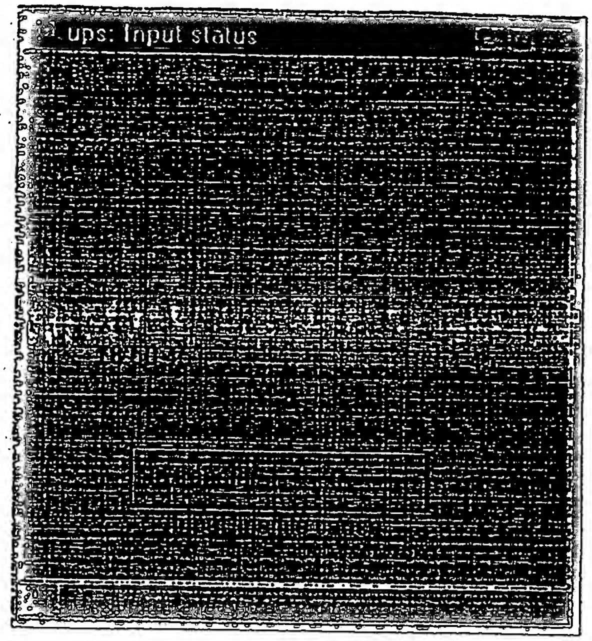

Fig. 6. The input line status panel.

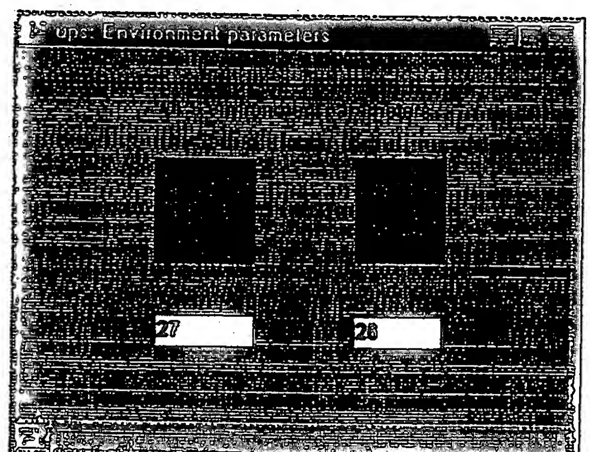

Fig. 7. The environment parameters.

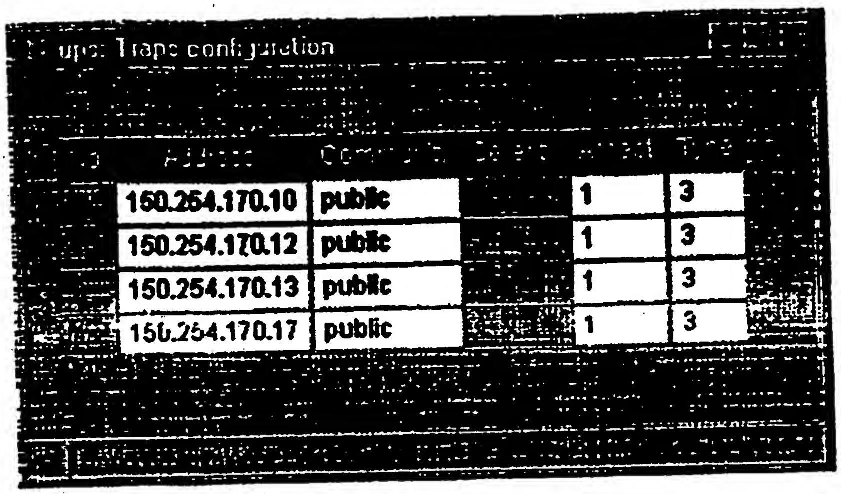

Fig. 8. The trap configuration.

two objects allow the administrator to recognize easily the UPS's working status.

On the right of these two objects there are two subpanel objects that open two windows: Battery Status (Fig. 5) and the Input Line Status (Fig. 6).

In the Battery Status window there are two text fields presenting the elapsed time when in On-Battery status and the estimated remaining time until the full battery runs out, which is calculated based on the prior UPS calibration. Below the text fields there are two bars presenting the battery capacity (green) and the UPS load (red). If the battery capacity falls below 30% an alarm is sounded.

The Input Line Status panel presents the UPS input voltage on the chart. Below it is the switch presenting, in a text form, the last cause of failure.

At the bottom of the main management panel there are four subpanels that open the Environment

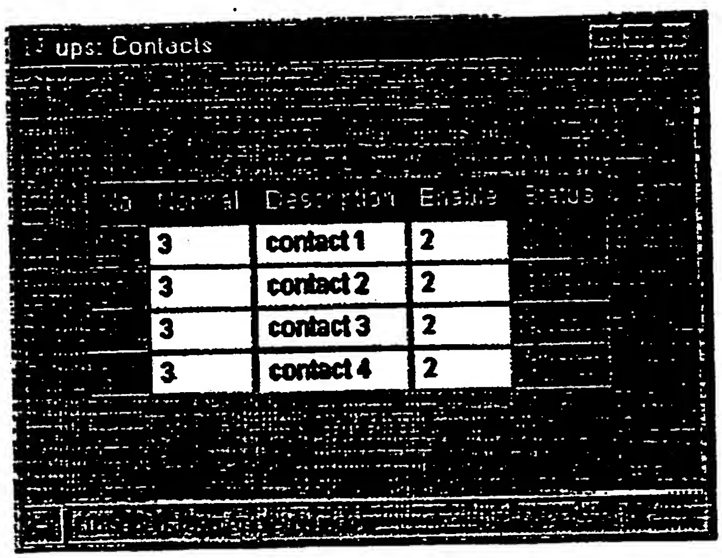

Fig. 9. The contacts configuration.

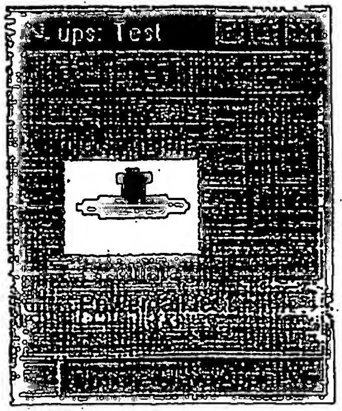

Fig. 10. The test panel.

Parameters panel, (Fig. 7), the Trap Configuration panel (Fig. 8), the Contacts Configuration panel (Fig. 9) and the Test panel (Fig. 10).

In the right bottom corner there is the trap object — all received traps from this UPS. Thus, the administrator is informed about all events on the managed UPS.

The Environment Parameters panel consists of two bars and two related text fields that present the value visualized by the bars. These objects present the temperature (red) and relative humidity (blue) of the UPS environment. If the temperature exceeds 35°C or the humidity exceeds 65% an alarm is sounded.

The Trap Configuration panel includes only the table presenting defined trap receivers.

The Contacts Configuration panel also includes the table with the configuration of the contacts on the APC SNMP adapter.

The last panel, the Test panel, includes only one object allowing the user to perform the simulation of a power failure. Marking the positions *Simulate* and pressing the *OK* button will simulate a power failure. This action will cause traps to be sent that will trigger an alarm.

Most of the above-described objects were predefined as templates in the *Toolbar* window (Fig. 11). These facilities define similar panels for future users.

In this panel from the left side there are:
- Trap receivers
- Ambient temperature
- Ambient humidity
- Contacts table
- UPS model name
- Time when in On-Battery state
- Battery capacity
- Remaining time when in On-Battery state
- Input voltage chart
- Input line last fail cause
- UPS status
- UPS load
- Simulate power failure
- On-line state indicator

The system of making management services available by the Web interface allows the administrator to manage SNMP devices with the Web browser. The possibility of defining his or her own management panels, which visualize chosen SNMP objects, opens a new opportunity to manage the Web interface.

Up to now, in the available management systems based on Web technology, the management domain consisted of a set of controlled devices. It was not possible to easily visualize SNMP variable mirroring device specific parameters, like ambient temperature, link capacity or link error status. WebMan has such ability.

The implemented system has many advantages, but it can still be developed further. It is necessary to create a virtual grid so it is easier to position object and a MIB browser would be very useful to make it

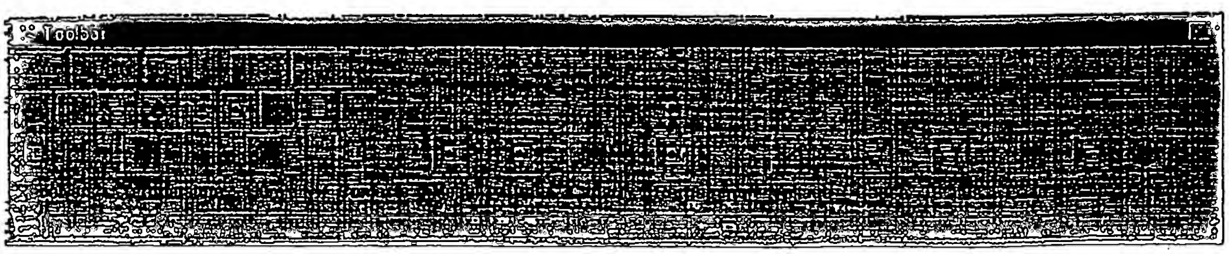

Fig. 11. The Toolbar window with templates for UPS management.

easier to enter OIDs (Object IDs). However the most important development work is to create more WebMan objects that can visualize SNMP objects in different ways.

5. Conclusion

This paper describes a new approach for management services based on the use of a Web browser to access MIB objects. It compared the features of Web-based access with standard access from management platforms or applications.

The implementation of the WebMan management system is presented. A characteristic feature of this system is the possibility for the user to specify their own management panel for the visualization of parameters related to managed devices. Communication between users and the server are implemented in a secured way by the use of encryption and user authentication.

As an example, the use of the WebMan system to manage a UPS system was presented. This example described how all the parameters accessible by objects in the MIB could be visualized for the user.

Future work it is expected to make an implementation of the system which cooperates between the Web server and Web browsers produced by various suppliers. Other planned modifications include:

- Implementation of algorithm secured transmission encoded with a public key.
- Extension of the number of graphic objects in the management panel.
- Review of the MIB tree.

Maciej Jarkowski (M.Sc.), born in 1974, studied computer science at the Poznań Technical University from 1993 to 1998. Between 1996 and March 1999 he worked for the Poznań Supercomputing and Networking Center. During this employment he implemented systems for network management.

Artur Binczewski received an M.Sc. degree in Computer Science from the Poznań University of Technology in 1993. His research interests concern computer networks, routing, multicasting and management. He is the Manager of Network Division at the Poznań Supercomputing and Networking Center.

Maciej Stroinski received a Ph.D. degree in Computer Science from the Technical University of Gdansk in 1987. Currently he is technical Director of the Poznań Supercomputing and Networking Center. He is also a lecturer in the Institute of Computing Science at the Poznań University of Technology. His research interests concern computer networks protocols and management. He is author or co-author of over 100 papers in major professional journals and conference proceedings.